

Description

METHOD FOR REDUCED ELECTRICAL FUSING TIME

BACKGROUND OF INVENTION

[0001] Technical Field:

[0002] The present invention relates generally to semiconductor devices and in particular to electrical programming of on-chip fuses of semi-conductor devices. Still more particularly the present invention relates to reducing fusing time during electrical programming of on-chip fuses of semiconductor devices.

[0003] Description of the Related Art:

[0004] Current semiconductors and other similar devices are often manufactured with a large number of fuse elements which are used to invoke the replacement of failing cells or components, and provide DAC trimming, etc. During testing of the device, selected fuses are usually blown by either a laser beam or an electrical current, depending on

the design of the fuse/device. In an electrical fuse design, electronically programmable fuses are blown by passing a current through the fuse link. The electrical current then causes a permanent change to the resistance of the fuse. The fuses that are blown are selected via one or more programming methods, which are generally known to those skilled in the art.

[0005] Figure 1 illustrates a prior art schematic of a portion of a device with fuse elements (eFuse circuits) and input signals/logic. Only two fuses are illustrated, although a complete device may comprise a much larger number of fuses. As shown, the electrical fuse circuits are fabricated with two latches per fuse. The first latch, which is called the fuse sense latch (or fuse latch) 103 is utilized to read the state of the fuse and is also utilized during the fusing process to enable/disable the blowing of the associated fuse. The second latch, called the pattern latch 113, is utilized to store the redundancy solution that was calculated for the device. According to Figure 1, the fuse latch 103 is in the "upper" register and the pattern latch 113 is in the "lower" register.

[0006] As illustrated, the two chains of latches (fuse latches 103 and pattern latches 113) are connected in serial fashion,

with each latch receiving its shift input from the previous latch and sending its shift output to the next latch in the chain. With this topology, enabling an electrical fuse to be programmed requires that both the output of the fuse latch 103 and pattern latch 113 corresponding to a specific fuse be set high (utilizing an AND logic function 105). Prior to fuse programming, the FSOURCE input 109 is set to a voltage high enough to program the fuses. Other inputs, including clock inputs ACLK 110 and BCLK 111, for example, along with input logic 116–119 are utilized to set the values within the latches. During the time that both latches are set high and global input GATEN is high, the programming transistor 107 for the eFuse turns on, blowing the connected fuse 106.

[0007] A major cost adder to the production of semiconductor devices (such as ASIC --application-specific integrated circuit-- logic chips) is the time and financial costs incurred when testing the device. A major portion of the testing and associated costs involves the fusing process (i.e., the process by which the fuses are selected and electrically blown). Current electrical fusing times are on the order of 200us per fuse. Due to the fusing current and on chip routing of relatively high voltage and high current

busses, fusing multiple fuses at a time is not a practical solution for typical semiconductor devices. With fusing current as high as 15mA per fuse, multiple simultaneous fuse blow becomes difficult and requires more device I/O's to be dedicated to the fuse blow input (called FSOURCE) supply. Using many input pads on a chip/package to support electrical fusing is not a cost effective solution. Thus, fuses are currently blown in a serial manner (i.e., one at a time) during testing because of the high fusing current and other factors described. With electrical fuses it is desirable to reduce fusing time as much as possible.

[0008] The fusing process can require a significant amount of time compared to the overall testing time. This is mainly due to the microsecond time scale used in fusing versus the nanosecond timings used for test. The fusing time is controlled by the rate at which a single '1' is clocked or shifted through the fuse latches 103. Only one fuse latch 103 is set high at a time, guaranteeing only one fuse 106 is "blown" at a time. With this design, fusing time required for one fuse is spent on every latch in the fuse latch shift chain.

[0009] For example, assume the fusing process takes 1 clock cy-

cle to blow each fuse. Then, the number of total clock cycles required for completing the test when the fuses are serially blown equals at least the number of fuses in the device. With the large number of fuses in current devices, typically hundreds or thousands, this serial implementation requires significant time commitment and is extremely inefficient for devices that only require a small percentage of the available fuses to be blown.

[0010] Another primary concern with current designs is the area overhead incurred due to the need to individually address electrical fuses on-chip. U.S. patent 6,426,911 provides one solution that addresses this area overhead problem by creating a shift register by a serial in/serial out wiring of the fuse latches and the connection of the gates of the blow transistors to the serial out port of each fuse latch. However, the problem of fuse time remains as the technique provided by the patent fails to reduce the one-by-one propagation of the signal through each latch of the fuse scan chain. That is, the programming information is propagated from latch to latch as a single "1" acting as a pointer that activates only one fuse latch per clock cycle. Also, only one fuse is blown per clock cycle to prevent the problems associated with high electrical current and other

overhead issues described above.

[0011] From the perspective of a scan path, particular types of latches are provided to enable a scan chain evaluation for the device. In Figure 1, the fuse scan chain connections are illustrated with LSSD (level sensitive scan design) latches. The fuse latches 103 and pattern latches 113 are serially connected and wired into the fuse logic. The fuse pattern is loaded into the lower or pattern register prior to programming. The upper or fuse register has a single bit ("1") loaded and "walked" down the shift register. The clock frequency utilized to clock the fuse register gives simple control over the fuse programming time.

[0012] As illustrated by Figure 1, each serial register of latches (LSSD scan latches) comprises a dual latch. The first latch in an LSSD scan latch is called the L1 and is loaded with clock signal ACLK 110. The second latch in the LSSD scan latch is called the L2 and is loaded with signal BCLK 111. When the fuse 107 is sensed, the value is sensed into the latch L1. Sensing of the fuse is done with a single ended resistor divider technique that is generally known to those skilled in the art. If the fuse 107 is intact, the latch will be pulled low by the connection from the fuse to FSOURCE 109, which must be at 0 Volts (GND) during sensing of the

fuse. If the fuse 107 is programmed (blown), the latch L1 will remain high. Once the signal that enables fuse sense goes low, the fuse latch sensing is complete.

[0013] The present invention recognizes the inefficiencies that exist in the design and testing of current devices and recognizes that it would be desirable to reduce the time and costs associated with the fusing process, particularly when only a small fraction of the total number of fuses are actually being blown. A method and device that enables efficient completion of the fuse blowing stage of device testing would be a welcomed improvement. It would be further desirable if the method and device also enabled reduced area overhead along with the reduced fusing time. These and other benefits are provided by the invention described below.

SUMMARY OF INVENTION

[0014] Disclosed is a method and electrical fuse circuit design for reducing the testing time for a semiconductor device which uses fuse circuitry. A two-to-one multiplexer (MUX) is provided at each fuse circuit in addition to the fuse latch and pattern latch associated with the fuse circuit. Information on which fuse is to be blown is stored in the fuse's pattern latch. Utilizing the output generated by the

pattern latch, the MUX then allows the shifted "1" to either go only to the current latch in the shift chain (program state) or go to the current latch and the next latch in the shift chain (bypass state). Any number of MUXes in a row could be in the bypass state causing the walking "1" to follow the bypass paths of the MUXes until it reaches a latch and MUX in the program state. Since the AND logic to program a fuse requires the fuse and pattern latch be set, only the latch that is in the program state will program the fuse. The fuse latches in the bypass state also load a "1" but since their corresponding pattern latches are not set, the AND logic will not program the associated fuses. Accordingly, rather than serially shifting through each fuse latch within the device one at a time, the invention enables the shifted "1" to go directly to the next fuse to be programmed.

[0015] According to one embodiment, a '1' is utilized in the pattern latch to specify that a fuse is to be blown and a "0" specifies that a fuse is not to be blown. The complement output of the pattern latch is ANDed with a control signal to select the MUX state from among an upper (latch) input and a lower (MUX bypass) input. The control signal is asserted to a high state during fuse blow. If the pattern latch

output indicates the fuse is to be blown, the high (1) input of the MUX is selected and the shift path is routed through the fuse latch. When the pattern latch output indicates the fuse is not to be blown, the low (0) input of the MUX is selected, which in effect results in a bypass of the latch. This latter operation bypasses the fuse latch and sends the walking "1" downstream to the next latch of the sequence of latches (or eFuse circuits) at which the pattern latch indicates the fuse is to be blown.

[0016] Since only a small percentage of the total number of fuses is to be blown during testing (e.g., 10 percent of the total number of fuses), a substantial time-savings is provided (i.e., almost 10X improvement for the example) by the invention with minimal circuit overhead. When serial read-out of the fuse latches is required, the control signal is held low. This forces all the MUXes to select the high input and also forces all of the fuse latches to be in the shift path regardless of the state of the pattern latches. This forced non-bypass state can also be used for testing of the fuse latches where each latch needs to be in the shift path. Implementation of the invention thus reduces processing time for traditional eFuse circuit devices and testing procedures thereof.

[0017] The above as well as additional objectives, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF DRAWINGS

[0018] The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

[0019] Figure 1 is a schematic diagram of a prior art fuse latch circuit that is tested utilizing prior art methods with long testing times;

[0020] Figure 2 is a schematic diagram of a fuse-latch circuit that selectively programs electrical fuses via a latch bypass mechanism to provide timing benefits when testing the device according to one embodiment of the present invention;

[0021] Figure 3A is a logical diagram of a series of multiplexers (MUX) with their input and select signals within an eFuse circuit device according to one embodiment of the present invention;

- [0022] Figure 3B is a table illustrating the time difference in clock cycle between a traditional fuse latch shift process and the bypass process implemented by the MUXes of Figure 3A according to one embodiment of the invention; and
- [0023] Figure 4 is a basic flow diagram illustrating the process by which only fuses to be blown are selected during a walk-through of a device designed according to one embodiment of the invention.

DETAILED DESCRIPTION

- [0024] The present invention provides a novel method and device design for reducing electrical fusing times in semiconductor devices during testing. The invention comprises a circuit design and methodology for completing more efficient testing of electrical fuses within a semiconductor device by selectively walking the blow signal to only those fuses that are to be blown.
- [0025] Referring now to Figure 2, there is illustrated a schematic circuit diagram of a semiconductor device (eFuse circuits) designed to implement selective fusing operations according to the methods of the present invention. Specifically, Figure 2 shows the new fuse latch circuitry to enable the novel selective fuse blowing and bypass features. Device 200 comprises many of the similar logic and other

components of Figure 1, including serially connected fuse circuits (see box labeled "Repeated Circuits for 1 Fuse"). Consequently, like numerals are utilized for the signal inputs and logic and portions of the initial fuse circuit connected thereto. Corresponding numbers are utilized to differentiate the components (both old and new) of the repeated circuits to indicate that even though similar fuse and latch circuitry are being utilized, these components are connected to additional components and provide a different overall functionality than similar components in Figure 1. In Figure 2, two additions are made to the eFuse circuits from previous designs. First, a MUX is added to facilitate "skipping" fuse blow time (or "bypassing" fuse latches) on fuses that are not to be blown. Second, AND gate (or logic) is provided to enable correct selection of one of the inputs to the MUX.

[0026] Each fuse circuit 220 comprises a fuse 206, which may be blown when current signals are provided to a source and gate of transistor 207. Each eFuse circuit 220 further comprises fuse sense latch or fuse latch (FLO) 203, utilized to read the state of the fuse and during the fusing process to enable the fuse that is currently being blown. Also, each eFuse circuit 220 comprises pattern latch (PLO)

213, utilized to store the fuse solution previously calculated and programmed for the device.

[0027] As mentioned above, in addition to these components, eFuse circuit 220 comprises a two-to-one" multiplexer (MUX) 221 (also labeled M0 to indicate the first of the series of MUXes, M0 to MN, where N is an integer) and an AND logic 225 (also labeled A0 to indicate a first in a series, A0 to AN), whose output serves as the select signal for MUX 221. MUX 221 is utilized to allow the shifted '1' or walking '1' to either go to the next fuse latch in the shift chain or bypass (from a timing perspective) the next fuse latch (or latches) in the shift chain depending on whether the corresponding fuse is to be blown. The first (or upper) MUX input is coupled to the true output (L2T) of the previous fuse latch 203 and is a low (0) input. The second (or lower) MUX input is coupled to the previous MUX 221 in the series of eFUSE circuits and is a high (1) input.

[0028] Notably, although the illustrative embodiments assigns the upper input of MUX 221 to the true output of the fuse latch (L2T) and the lower input of MUX 221 to the component output of the pattern latch (L2C), the invention may likewise be implemented with the inputs assigned in the

reverse manner. That is, a reverse configuration of MUX inputs may be implemented, but with some additional change to the AND logic and/or the pattern latch. With a reversed-input MUX configuration, the AND logic is replaced by a NAND gate and/or the output from AND logic is passed through an inverter. Other implementations are possible and all fall within the teachings of the present invention.

[0029] Figure 2 also illustrates possible inputs (signals and logic) to the device. Input circuitry and signals 210 provide the inputs required from powering and running/operating the eFuse circuits 220. Input circuitry and signals 210 include multiple clock signals, ACLK, BCLK, CCLK, gating logic, and a power source FSOURCE. Specific implementation of the input circuitry and signals are not particularly important for the invention and conventional means may be utilized. However, in one embodiment of the invention, a specific configuration of input signals are provided for additional functionality, and this configuration of input signals is described within co-pending U.S. Patent Application, Serial No. XXX (Attorney Docket Number BUR920030088, entitled "Method for Splitting Shift and Scan Paths on Scan Only LSSD Latches" and filed on June

XX, 2003. The relevant content of that application is hereby incorporated by reference.

[0030] The information on whether a fuse is to be blown is programmed and stored in pattern latch 213, which is local to each eFuse circuit 220. According to the invention, pattern latch 213 provides an output that indicates when the associated fuse is to be blown and sends that output to MUX 221. For the illustrative embodiment, pattern latch 213 is programmed with a '1' when the associated fuse 206 is to be blown, and pattern latch 213 is programmed with a '0' when the fuse 206 is not to be blown.

[0031] In the illustrative embodiment, the complement output (i.e., the output of port L2C) of PL0 213 is utilized as one input to AND logic 225. AND logic 225 also receives a program input, EFUSEPROGRAM (EFP) 223, and both EFP 223 and the complement of the value stored/programmed in PL0 213 are ANDed together to select one of the two inputs to MUX 221. The upper input of MUX 221 is the input corresponding to the serial flow of the shifted "1" through the fuse latches (i.e., L2T) that in certain circumstances results in the fuse 206 being blown. The lower input, however, operates as a bypass trigger for the eFuse circuit so that the fuse latch is not triggered to initiate

programming of the fuse. Rather, the shifted 1 is passed on to a next sequential MUX of a next sequential eFuse circuit and so on. This process is also illustrated by the flow chart of Figure 4B, which is now described.

[0032] Figure 4 provides a general summary of the processes completed by eFuse circuits 200 in completing a test procedure that involves a blowing of some fuses within a device and bypassing of other fuses. Beginning at block 421, the pattern latch is first programmed by any one of the known methods for programming the pattern latch, and the pattern latch provides a complement output (L2C). The L2C output is ANDed with the EFP input as shown at block 423. A shifted 1 arrives at the eFuse circuit as indicated at block 425, and a determination made at block 427 whether the MUX select signal from the AND logic selects the fuse latch input (over the MUX bypass input).

[0033] If the MUX selects the fuse latch input, a fuse blow process is indicated for the eFuse circuit as shown at block 431, and fuse blow process is completed before the shifted 1 is passed to the next eFuse circuit. However, when the MUX does not select the fuse latch input, the MUX bypass input is selected and the shifted 1 is sent to the next eFuse circuit as shown at block 429. The processing

at the current eFuse circuit then ends, as shown at block 433.

[0034] Figure 3A provides a close-up view of six serially-connected MUXes (M0 through M5) and their inputs and select signals according to one embodiment. As shown, M0 and M4 are both triggered to select their upper inputs from the preceding fuse latch. The triggering is completed by the AND logics and based on the L2C outputs from the corresponding pattern latches, respectively.

[0035] With reference again to Figure 2 and Figure 3, where appropriate, EFP 223 is asserted to a high state during fuse blow. If the L2C output of latch PL0 213 is low (i.e., the pattern latch L2 is high and the fuse is to be blown), then the upper input of MUX 223 coupled to the fuse latch (i.e., the input that indicates the fuse latch is to initiate the fuse blow process that ultimately blows the fuse 206) is selected, and the shift path is routed through latch FL0. This means the "walking 1" that enables fuse blow in the fuse latch chain is sent through latch FL0 to initiate a fuse blow. Consequently, when EFP 223 is '1' and L2T from PL0 is '1' and L2T from FL0 is '1', the programming transistor (T0) 207 turns on, blowing the fuse 206.

[0036] If the PL0 latch 213 is set low, the corresponding L2C out-

put is high and the MUX 221 chooses its lower input corresponding to the bypass input that is received from the previous MUX. The FL0 is bypassed (i.e., time to switch through latches, etc. does not hold up the propagation of the walking "1" to the next eFuse circuit) and the walking '1' is sent downstream to the latches where the L2C output from the pattern latch is high. All fuse latches that have a corresponding pattern latch set low (L2C = "1") still load the walking "1" but the fuse does not blow due to the pattern latch L2T output that goes to the AND logic. Since L2T is low for a fuse that is not to be blown, the programming device (A1) cannot turn on. Accordingly, the MUX logic allows the number of "walking 1" shift cycles to equal the number of fuses to be blown from a timing perspective.

[0037] Figure 3B is a table showing the response of the MUX logic of Figure 3A to the select inputs of the AND logic and the clock time required for passing the shifted 1 through the sequence of eFuse circuits. The MUXes are indicated in each column and the parameter being tabulated is shown in their respective rows, each entry now corresponding to the particular MUX. For illustrative purposes, it is assumed that a single clock cycle is required to complete a fuse

blow operation at any one of the eFuse circuits at which the blow operation is completed. A bypass of the fuse blow operation is assume to be almost negligible in terms of time (i.e., 1/100 of the time for the fuse blow). For a fuse with MUXes in bypass preceding it, the fusing time for that fuse will be reduced by the propagation delay of the MUXes. This however is negligible since the MUX delays are less than 1/4 nanosecond each and the fusing time is 200 microseconds.

[0038] The pre-programmed pattern latch outputs are illustrated in the first row with a 1 indicating that the fuse is to be blown and a zero indicating the fuse should not be blown (i.e., bypassed). Pattern logic of eFuse circuits 1 and 4 are programmed to blow their fuses. Accordingly, the second row indicates which logic component is triggered/selected by the MUX during the serial shifting of the shifted 1 through the eFuse circuits. Again fuse latch input is selected for both eFuse circuits 1 and 4, while the MUX (bypass) input is selected for all other eFuse circuits.

[0039] With respect to the time (clock cycles) required to complete each operation at each circuit, row 3 provides the 1 to 1 ratio of clock cycles to number of circuits as required by the prior art methods. Row 4 provides the clock cycles

based on the novel methodology and circuit design of the invention. As can be seen, only the fuse blow process of circuits 1 and 4 utilizes the full clock cycle to complete blow operation at those eFuse circuits. The other circuits are bypassed and do not require a time of processing the full clock cycle before proceeding to the next circuit as in the prior art.

[0040] During serial readout of the fuse latches (i.e., where all the fuse latches are required to be in the shift path), EFUSEPROGRAM is held low, forcing all MUXes 221 to choose the upper input (from the fuse latches). This also forces all fuse latches to be in the shift path regardless of the state of the pattern latches. The pattern latch register thus controls the "effective length" of the fuse latch register to equal the number of '1's onset latches in the pattern register only when EFUSEPROGRAM is high.

[0041] One advantage of the method implemented by the invention is that "one at a time" fusing is maintained, requiring only one FSOURCE input signal for many eFuse circuits. However, rather than spending 200μs at every eFuse circuit location (as with prior art), the invention enables the 200μs be spent only on those fuses that are to be blown. Thus, for example, if only 10 percent of the fuses are to

be blown, a blow operation completed during testing of a device designed according to the invention achieves a 10x improvement in fusing time over the previous serial methods. More importantly, this 10x improvement comes without additional power requirement and with very little circuit overhead. The fusing design is a time-efficient blowing method that substantially reduces by several magnitudes the testing time for the device.

[0042] While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.